**In the Specification**

**Please amend as follows:**

Page 4, lines 3-16

Two or more GCR functions are loaded into GCR processor 14 (shown referenced as GCR-1, GCR-2, …, GCR-*N*). For N GCR ~~*N* GCR~~ functions, information 16 to be transmitted is converted therein into *N*-ary form. The *N-ary* form is one wherein the information at each pixel can be represented by one of N levels ~~*N* levels~~. For example, if two GCR functions are to be used (*N*=2), the *N*-ary form for the information is binary, so that each pixel is represented by one of two levels [0, 1]. One might choose GCR1 ≡ 100%GCR (maximum amount of gray component is replaced with K) to encode every 0, and GCR2 ≡ 0%GCR (no K is used to replace the gray component) to encode every 1. Expressed analytically, GCR1 and GCR2 of this simple example perform the transformation: $CMYK_1$ = (C-min(C,M,Y), M-min(C,M,Y), Y-min(C,M,Y) , min(C,M,Y)); for GCR-1 ≡ $GCR_{100\%}$; $CMYK_2$ = (C, M, Y, 0); and for GCR-2 ≡ $GCR_{0\%}$; where C, M and Y on the right side of the equations indicate values into the GCR processor 14, and C, M, Y, and K on the left side of the equations indicate values output from GCR processor 14.

Page 7, line 21 to Page 8, line 7

In the alternative, spatial properties of the high-frequency image are exploited to determine $K_H$. Assuming halftoning with rotated screens, CMY dots do not frequently overlap completely in the highlight and mid-tone regions. Typically, the amount of overlap of CMY dots covers a small area percentage at highlights and mid-tones. If the input image has perfect dots and a scanner that has high enough resolution that it is able to perfectly scan the dots and their overlaps is used, then the scanned RGB values thereof only assume a small number of combinations that depend on the geometry of the dots against the scanner resolution. If each of the RGB values for any given pixel in the scan are low enough then that pixel essentially comprises all black color. A black-colored pixel can then be assumed a ~~K-dot~~ K dot. This ~~K-dot~~ K dot assumption is a good approximation because for low RGB there must be either a ~~K-dot~~ K dot or an overlapping of CMY dots ~~CMY-dots~~. Because the overlap area of CMY dots ~~CMY-dots~~ is often small in highlights and mid-tones (due to rotated screens) it is therefore probable that the black-colored pixel is a ~~K-dot~~ K dot. Additional information, for example, ensuring that the pixel has very low chroma, is also useful. Preferably, a combination of luminance and chroma can be used to distinguish between a K dot and an overlap of CMY dots.

Page 8, lines 8-25

Additionally, at the dot center, a small K dot on a light background will have a lightness that is higher than a larger K dot on a darker background. Thus a transformation is necessary to normalize the lightness of the dot based on the scanned RGB. This is done as follows. One out of ~~*N* GCRs~~ N GCRs can be estimated by first, in the training session, determining one image or region that was processed with each GCR. For each region or image, compute $\beta i, K_L, L = E(K_H | K_L, L, GCR=i)$ ~~□i, K_L, L□E(K_H | K_L, L, GCR=i)~~, i.e., the average value of $K_H$ for each ($K_L$, L) pair. In the on-line detection phase, find the $\beta$ that ~~□that~~ is the closest to the received $K_H$. In the binary case, there are only two values to compute, $\beta(1, K_L, L)$ and $\beta(2 K_L, L)$ ~~□1, K_L, L and □□. K_L, L~~ so that it is easier to set up threshold: $\tau(K_L, L) = [\beta(1, K_L, L) + \beta(2 K_L, L)]^{1/2}$ such that $K_H$ is simply compared to a threshold $\tau(K_L, L)$ ~~□K_L, L) = [□1, K_L, L + □□. K_L, L□½ such that K_H is simply compared to a threshold □K_L, L)~~. The array of the thresholds (typically 256x256) is preferably setup beforehand to simplify the computation, i.e., detection can be made with one look-up and one comparison. Thus the determination of which pixel is associated with a K dot depends on the average lightness of the region. Because $K_L$ relates to RGB in an average colorimetric way and does not comprehend microscopic halftone dot geometry, it can be reduced (using *S* as defined) to: $K_L = 1 - \max(S(R), S(G), S(B))$. The luminance value from the low-resolution scans is: $L = a\,S(R) + b\,S(G) + c\,S(B)$.

Page 8, line 26 to Page 9, line 19

In the alternative, for decoding the GCR-based watermark, the function β(i, RGB) which ~~□i, R, G ,Bwhich~~ relates the expected amount of K to the scanned RGB signal, is obtained through a separate calibration process, as follows. A set of CMY values is generated, preferably as a 3-dimensional grid in CMY space. These values are processed through each of the ~~*N* GCR~~ N GCR functions to obtain *N* sets of CMYK values. If other functions, such as ink-limit, and tone reproduction curves (TRCs), are applied during nominal printing of images, these functions are to also be applied to the CMYK data. Subsequently, from each CMYK data set, a target of patches is generated, printed and scanned. The estimate β(i, RGB)=1,... N ~~□i R,G,Bi=1,...,N~~, of the amount of black colorant in each patch is obtained for each of the N scanned targets corresponding to the ~~*N* GCR~~ N GCR functions. One exemplary method for doing this has been described previously. The black estimates are then used to populate a 3-dimensional lookup table (LUT) whose inputs are the scanned RGB values, and whose outputs are the *N* values of β ~~of □~~ corresponding to the N GCR ~~*N* GCR~~ functions.

Given a scan of an arbitrary printed image, the black estimate $K_H$ is derived using the same algorithm employed in the aforementioned calibration procedure. The RGB values of the scanned pixels are used to index the 3D LUT, and $K_H$ is compared with each of the *N* number of β values ~~□ values~~. The closest β value ~~□ value~~ determines which of the *N*-symbols was encoded at the given image location.

In the case where *N*=2, one can alternatively fill the 3-D LUT with the threshold: τ(RGB) ~~□ R, G, B)~~ = $[\beta(1, RGB) + [\beta(2, RGB)]^{1/2}$ ~~[□1, R, G, B+ □□ R, G, B□½.~~ The received black estimate $K_H$ is then simply compared with τ(RGB) ~~□R, G, B)~~ to determine which of the two symbols was encoded at the given image location.

Page 9, line 20 to Page 10, line 2

It is to be appreciated that methods for filling 3-D LUTs given an arbitrary sampling of input and output data is well known in the art. See for example: R. Bala, DEVICE CHARACTERIZATION, Digital Color Imaging Handbook, Chapter. 5, by: G. Sharma, Ed., CRC Press, 2003. It is also noted that pre-conditioning transforms may be applied prior to the 3-D LUT that increase the efficiency and accuracy of the 3-D LUT. For example, a 3x3 pre-conditioning matrix that approximately maps scanned RGB to printer CMY can be followed by a 3-D LUT that maps the said printer CMY to the β values ~~□ values~~. Since the original calibration targets are designed as grids in CMY space, such a preconditioning transform should result in a better utilization of nodes in the 3-D LUT. Also it attempts to explicitly recover the original GCR, which is a mapping from CMY to K, rather than a mapping from scanner RGB to K.